AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method, comprising:

installing a program onto a target machine, the program having intermediate representation;

executing the program using the intermediate representation and an initial profile data; ~~and~~

generating a current profile data;

comparing the current profile data with the initial profile data; and

~~responsive to a change in profile data collected while the program executes which exceeds a predetermined threshold, recompiling the program while the target machine is idle.~~

recompiling the intermediate representation to optimize the program when the current profile data in comparison with the initial profile data has exceeded a predetermined threshold.

2. (Currently Amended) The method of claim of claim 1, wherein ~~said~~ installing further comprises:

installing a continuous compiler;

installing a runtime monitor;

copying ~~an~~ the intermediate representation ~~of a source file~~ to the target machine;

building ~~a~~ the initial profile ~~database~~ data; and

compiling the intermediate representation to create an executable file.

marking the branch as inverted.

11. (Currently Amended) A ~~computer readable~~ machine-readable medium having stored thereon data representing sequences of instructions, the sequences of instructions which, when executed by a ~~processor~~ machine, cause the ~~processor~~ machine to ~~perform the steps of~~:

installing a program onto a target machine, the program having intermediate representation;

executing the program using the intermediate representation and an initial profile data; ~~and~~

generating a current profile data;

comparing the current profile data with the initial profile data; and

~~responsive to a change in profile data collected while the program executes which exceeds a predetermined threshold, recompiling the program while the target machine is idle.~~

recompiling the intermediate representation to optimize the program when the current profile data in comparison with the initial profile data has exceeded a predetermined threshold.

12. (Currently Amended) The ~~computer readable~~ machine-readable medium of claim of claim 11, wherein ~~said~~ installing further comprises:

installing a continuous compiler;

installing a runtime monitor;

copying ~~an~~ the intermediate representation of ~~a source file~~ to the target machine;

building ~~a~~ the initial profile ~~database~~ data; and

compiling the intermediate representation to create an executable file.

13. (Currently Amended) The ~~computer readable~~ machine-readable medium of claim 11, wherein ~~said~~ executing further comprises:

running an executable version of the program;

collecting samples of process information at a controlled rate; and

while the target machine is idle, generating binary level and high level profiles.

14. (Currently Amended) The ~~computer readable~~ machine-readable medium of claim 11, wherein recompiling further comprises customizing compiler optimizations based on the current profile data generated during program execution.

15. (Currently Amended) The ~~computer readable~~ machine-readable medium of claim 14, wherein ~~said~~ customizing compiler optimizations is performed using annotations in a high level representation of an executable program which relate portions of the executable to the high level representation.

16. (Currently Amended) The ~~computer readable~~ machine-readable medium of claim ~~16, wherein creating said annotations comprises:~~ 15, wherein the sequences of instructions which, when executed by the machine, further cause the machine to create the annotations, wherein creating comprises:

creating a new action node;

assigning to the new action node a major ID from a precomputed ID;

3. (Currently Amended) The method of claim 1, wherein ~~said~~ executing further comprises:

running an executable version of the program;

collecting samples of process information at a controlled rate; and

while the target machine is idle, generating binary level and high level profiles.

4. (Currently Amended) The method of claim 1, wherein recompiling further comprises customizing compiler optimizations based on the current profile data generated during program execution.

5. (Currently Amended) The method of claim 4, wherein ~~said~~ customizing compiler optimizations is performed using annotations in a high level representation of an executable program which relate portions of the executable to the high level representation.

6. (Currently Amended) The method of claim ~~6, wherein creating said annotations comprises:~~ 5, further comprising creating the annotations, wherein creating comprises:

creating a new action node;

assigning to the new action node a major ID from a precomputed ID;

assigning a new action number to the new action node;

setting a previous action node pointer of the new action node to NULL;

marking a compiler phase in which the new node was created; and

marking an action of the new node as created.

7. (Currently Amended) The method of claim ~~6, wherein duplicating an annotation comprises:~~ 5, further comprising duplicating the annotations, wherein duplicating comprises:

creating two new action nodes;

copying a major ID to the new action nodes from an action node of instructions being copied;

assigning new action numbers to the two new nodes;

setting previous action node pointers of the new nodes to the action node being copied;

marking a compiler phase in which the nodes was duplicated; and

marking an action of the new nodes as duplicated.

8. (Currently Amended) The method of claim ~~6, wherein deleting an annotation comprises:~~ 1, further comprising deleting the annotations, wherein deleting comprises:

creating a new action node;

copying a major ID from an action node of an instruction being deleted to the new action node;

assigning a new action number to the new action node;

setting a previous action pointer in the new action node to the action node of the instruction being deleted;

marking a compiler phase in which the node was deleted; and

marking an action of the deleted node as deleted.

9. (Currently Amended) The method of claim ~~6, wherein merging of annotations comprises:~~ 1, further comprising merging the annotations, wherein merging comprises:

creating a new action node;

copying a major ID from a previous action node of instructions being merged;

assigning a new action number to the new action node;

setting a previous action pointer of the new node to a list of nodes of the instruction being merged;

adding the new action to a next actions pointer list of previous actions;

marking a compiler phase in which the node was merged;

marking an action of the new node as merged.

10. (Currently Amended) The method of claim ~~6, wherein annotation branch inversion comprises:~~ 1, further comprising inverting an annotation branch, wherein inverting comprises:

creating a new action node;

copying a major ID from a branch instruction being inverted;

assigning a new action number to the new node;

setting a previous action pointer of the new node to a node of a branch being inverted;

marking a compiler phase in which the inversion occurred; and

assigning a new action number to the new action node;

setting a previous action node pointer of the new action node to NULL;

marking a compiler phase in which the new node was created; and

marking an action of the new node as created.

17. (Currently Amended) The ~~computer readable~~ machine-readable medium of claim ~~16, wherein creating said annotations comprises:~~ 15, wherein the sequences of instructions which, when executed by the machine, further cause the machine to duplicate the annotations, wherein duplicating comprises:

creating two new action nodes;

copying a major ID to the new action nodes from an action node of instructions being copied;

assigning new action numbers to the two new nodes;

setting previous action node pointers of the new nodes to the action node being copied;

marking a compiler phase in which the nodes was duplicated; and

marking an action of the new nodes as duplicated.

18. (Currently Amended) The ~~computer readable~~ machine-readable medium of claim ~~16, wherein creating said annotations comprises:~~ 15, wherein the sequences of instructions which, when executed by the machine, further cause the machine to delete the annotations, wherein deleting comprises:

creating a new action node;

copying a major ID from an action node of an instruction being deleted to the new action node;

assigning a new action number to the new action node;

setting a previous action pointer in the new action node to the action node of the instruction being deleted;

marking a compiler phase in which the node was deleted; and

marking an action of the deleted node as deleted.

19. (Currently Amended) The ~~computer readable~~ machine-readable medium of claim ~~16, wherein creating said annotations comprises:~~ 15, wherein the sequences of instructions which, when executed by the machine, further cause the machine to merge the annotations, wherein merging comprises:

creating a new action node;

copying a major ID from a previous action node of instructions being merged;

assigning a new action number to the new action node;

setting a previous action pointer of the new node to a list of nodes of the instruction being merged;

adding the new action to a next actions pointer list of previous actions;

marking a compiler phase in which the node was merged;

marking an action of the new node as merged.

20. (Currently Amended) The ~~computer readable~~ machine-readable medium of claim ~~16, wherein creating said annotations comprises:~~ 15, wherein the sequences of

instructions which, when executed by the machine, further cause the machine to invert an annotation branch, wherein inverting comprises:

creating a new action node;

copying a major ID from a branch instruction being inverted;

assigning a new action number to the new node;

setting a previous action pointer of the new node to a node of a branch being inverted;

marking a compiler phase in which the inversion occurred; and

marking the branch as inverted.

21. (Currently Amended) A system, comprising:

a target machine having a storage device, the storage device having stored therein a routine for transparent continuous compilation; and

a processor coupled ~~to~~ with the storage device ~~which when executing the routine:~~, the processor to:

~~installs~~ install a program onto ~~a~~ the target machine, the program having intermediate representation and a continuous compiler;

~~executes~~ execute the program using the intermediate representation and an initial profile data; ~~and~~

generate a current profile data;

compare the current profile data with the initial profile data; and

~~responsive to a change in profile data collected while the program executes which exceeds a predetermined threshold, recompiles the program while the target machine is idle.~~

recompile the intermediate representation to optimize the program when the current profile data in comparison with the initial profile data has exceeded a predetermined threshold.

22. (Currently Amended) The system of claim of claim 21, wherein ~~said~~ installation further comprises:

installing a continuous compiler;

installing a runtime monitor;

copying ~~an~~ the intermediate representation ~~of a source file~~ to the target machine;

building ~~a~~ the initial profile ~~database~~ data; and

compiling the intermediate representation to create an executable file.

23. (Currently Amended) The system of claim 21, wherein ~~said~~ execution further comprises:

running an executable version of the program;

collecting samples of process information at a controlled rate; and

while the target machine is idle, generating binary level and high level profiles.

24. (Currently Amended) The system of claim 21, wherein recompilation further comprises customizing compiler optimizations based on the current profile data generated during program execution.

25. (Currently Amended) The system of claim 24, wherein ~~said~~ customizing compiler optimizations is performed using annotations in a high level representation of an executable program which relate portions of the executable to the high level representation.

26. (Currently Amended) The system of claim ~~26, wherein creating said annotations comprises:~~ 25, further comprising creating the annotations, wherein creating comprises:

creating a new action node;

assigning to the new action node a major ID from a precomputed ID;

assigning a new action number to the new action node;

setting a previous action node pointer of the new action node to NULL;

marking a compiler phase in which the new node was created; and

marking an action of the new node as created.

27. (Currently Amended) The system of claim ~~26, wherein creating said annotations comprises:~~ 25, further comprising duplicating the annotations, wherein duplicating comprises:

creating two new action nodes;

copying a major ID to the new action nodes from an action node of instructions being copied;

assigning new action numbers to the two new nodes;

setting previous action node pointers of the new nodes to the action node being copied;

marking a compiler phase in which the nodes was duplicated; and

marking an action of the new nodes as duplicated.

28. (Currently Amended) The system of claim ~~26, wherein creating said annotations comprises:~~ 25, further comprising deleting the annotations, wherein deleting comprises:

creating a new action node;

copying a major ID from an action node of an instruction being deleted to the new action node;

assigning a new action number to the new action node;

setting a previous action pointer in the new action node to the action node of the instruction being deleted;

marking a compiler phase in which the node was deleted; and

marking an action of the deleted node as deleted.

29. (Currently Amended) The system of claim ~~26, wherein creating said annotations comprises:~~ 25, further comprising merging the annotations, wherein merging comprises:

creating a new action node;

copying a major ID from a previous action node of instructions being merged;

assigning a new action number to the new action node;

setting a previous action pointer of the new node to a list of nodes of the instruction being merged;

adding the new action to a next actions pointer list of previous actions;

marking a compiler phase in which the node was merged;

marking an action of the new node as merged.

30. (Currently Amended) The system of claim ~~26, wherein creating said annotations comprises:~~ 25, further comprising inversing an annotation branch, wherein inversing comprises:

creating a new action node;

copying a major ID from a branch instruction being inverted;

assigning a new action number to the new node;

setting a previous action pointer of the new node to a node of a branch being inverted;

marking a compiler phase in which the inversion occurred; and

marking the branch as inverted.

31. (New) An apparatus, comprising:

a storage medium,

a processor coupled with the storage medium, the processor to:

install a program onto the target machine, the program having intermediate representation;

execute the program using the intermediate representation and an initial profile data;

generate a current profile data;

compare the current profile data with the initial profile data; and

recompile the intermediate representation to optimize the program when the current profile data in comparison with the initial profile data has exceeded a predetermined threshold.

32. (New) The apparatus of claim of claim 31, wherein installation further comprises:

installing a continuous compiler;

installing a runtime monitor;

copying the intermediate representation to the target machine;

building the initial profile data; and

compiling the intermediate representation to create an executable file.

33. (New) The apparatus of claim 31, wherein execution further comprises:

running an executable version of the program;

collecting samples of process information at a controlled rate; and

while the target machine is idle, generating binary level and high level profiles.

34. (New) The apparatus of claim 31, wherein recompilation further comprises

customizing compiler optimizations based on the current profile data generated during program execution.

35. (New) The apparatus of claim 34, wherein customizing compiler optimizations is performed using annotations in a high level representation of an executable program which relate portions of the executable to the high level representation.